DD2421 Machine Learning Lab 1

Yunsen Xing

September 10, 2025

Table of Contents

Table of Contents

Assignment - 0 - Content

- MONK-1: $(a_1 = a_2) \vee (a_5 = 1)$
- If attribute a_1 is equal to a_2, or $a_5 = 1$, then the sample belongs to the positive class.
- It is not determined by a single attribute.
- In order to determine the labels of the sample, we need three attributes, so the depth of the decision tree is 3.

Assignment - 0 - Content

Monk-2 is the most difficult to learn because all attributes are independent, and the decision tree must consider multiple combinations

- MONK-2: $a_i = 1$ for exactly two $i \in \{1, 2, \ldots, 6\}$
- Only when exactly two of the six attributes are equal to 1, this sample belongs to the positive class.
- It is not determined by a single attribute.
- In order to determine the labels of the sample, we need six attributes, so the depth of the decision tree is 6.

Assignment - 0 - Content

- MONK-3: $(a_5 = 1 \wedge a_4 = 1) \vee (a_5 \neq 4 \wedge a_2 \neq 3)$
- If (a_5=1 and a_4=1), or ($a_5 \neq 4$ and $a_2 \neq 3$), then it is positive.
- The dataset contains 5% noise, which makes it easy to overfit during learning, but only three attributes are involved.
- In order to determine the labels of the sample, we need three attributes, so the depth of the decision tree is 3.

Table of Contents

Assignment - 1 - Content

Entropy means unpredictability of a dataset

Definition

$\mathrm{Entropy}(S) = -\sum_i p_i \log_2 p_i$

Assignment 1

Dataset	Entropy
MONK-1	1.0000
MONK-2	0.9571
MONK-3	0.9998

Table of Contents

Assignment - 2 - Content

An example of entropy for a uniform distribution: rolling a coin:

- $P_{head} = 0.5; P_{tail} = 0.5$
- $\mathrm{Entropy(S)} = -\sum_i p_i \log_2 p_i = -0.5 \times \log_2^{0.5} -0.5 \times \log_2^{0.5} = 1$
- The result of a coin-toss has 1 bit of information

Assignment - 2 - Content

An example of entropy for a uniform distribution: rolling a die:

- $P_1 = \frac{1}{6}; P_2 = \frac{1}{6}; \ldots P_6 = \frac{1}{6}$
- $\mathrm{Entropy(S)} = -\sum_i p_i \log_2 p_i = 6 \times \left(-\frac{1}{6} \log_2 \frac{1}{6}\right) = -\log_2 \frac{1}{6} \approx 2.58$
- The result of a die roll has 2.58 bit of information.

Assignment - 2 - Content

An example of entropy for a uniform distribution: rolling a fake die:

- $P_1 = 0.1; P_2 = 0.1; \ldots, P_5 = 0.1; P_6 = 0.5$
- $\mathrm{Entropy}(\mathsf{S})$ = - $\sum_i p_i \log_2 p_i = -5 \times 0.1 \log_2^{0.1} - 0.5 \times \log_2^{0.5} \approx 2.1$
- A real die is more unpredictable (2.58 bit) than a fake (2.1 bit)

Assignment - 2 - Content

An example of entropy for a uniform distribution: rolling a fake die:

- $P_1 = 0.1; P_2 = 0.1; \ldots, P_5 = 0.1; P_6 = 0.5$
- $\mathrm{Entropy}(\mathsf{S}) =$ - $\sum_i p_i \log_2 p_i = -5 \times 0.1 \log_2^{0.1} - 0.5 \times \log_2^{0.5} \approx 2.1$
- A real die is more unpredictable (2.58 bit) than a fake (2.1 bit)

Assignment - 2 - Content

An example of entropy for a non-uniform distribution: rolling a fake die:

- $P_1 = 0.1; P_2 = 0.1; \dots, P_5 = 0.1; P_6 = 0.5$
- $\mathrm{Entropy}(\mathsf{S}) =$ - $\sum_i p_i \log_2 p_i = -5 \times 0.1 \log_2^{0.1} - 0.5 \times \log_2^{0.5} \approx 2.1$
- A real die is more unpredictable (2.58 bit) than a fake (2.1 bit)

Assignment - 2 - Content

- The greater the entropy $\Rightarrow$ The more evenly distributed $\Rightarrow$ The more difficult it is to predict
- The smaller the entropy $\Rightarrow$ The looser the distribution $\Rightarrow$ The easier it is to predict

Table of Contents

Assignment - 3 - Content

Using the provided function for calculating information gain:

```
for i in range(6):
    print(dtree.averageGain(m.monk1, m.attributes[i]))

for i in range(6):
    print(dtree.averageGain(m.monk2, m.attributes[i]))

for i in range(6):
    print(dtree.averageGain(m.monk3, m.attributes[i]))
```

Assignment - 3 - Content

Information Gain

Dataset	a_1	a_2	a_3	a_4	a_5	a_6
MONK-1	0.0753	0.0058	0.0047	0.0263	0.2870	0.00076
MONK-2	0.0038	0.0025	0.0011	0.0157	0.0173	0.0062
MONK-3	0.0071	0.2937	0.00083	0.0029	0.2559	0.0071

Assignment - 3 - Content

Conclusion:

- For the MONK-1 dataset, choosing attribute a_5 with an information gain of 0.2870
- For the MONK-2 dataset, choosing attribute a_5 with an information gain of 0.0173
- For the MONK-3 dataset, choosing attribute a_2 with an information gain of 0.2937

Table of Contents

Assignment - 4 - Content

Definition

$$\mathrm{Gain}(S, A) = \mathrm{Entropy}(S) - \sum_{k \in \mathrm{values}(A)} \frac{|S_k|}{|S|} \mathrm{Entropy}(S_k) \tag{1}$$

where S_k is the subset of examples in S for the attribute A has the value k.

For splitting, we choose the attribute that maximizes the information gain, because it provides the most reduction in uncertainty.

Table of Contents

Assignment - 5 - Content

Using the provided function for Building Decision Trees:

```
tree_monk1 = dtree.buildTree(m.monk1, m.attributes)
drawtree_qt5.drawTree(tree_monk1)
```

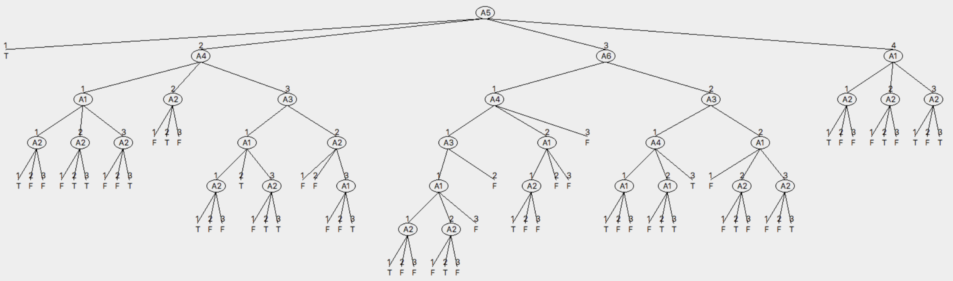
Compute the train and test set errors for the three Monk datasets for the full trees.

Dataset	Error training	Error testing
MONK-1	1.0 - 1.0 $=$ 0.0	1 - 0.8287 $\approx$ 0.1713
MONK-2	1.0 - 1.0 $=$ 0.0	1 - 0.6921 $\approx$ 0.3079
MONK-3	1.0 - 1.0 $=$ 0.0	1 - 0.9444 $\approx$ 0.0556

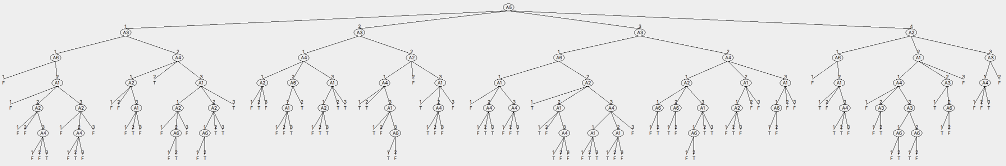
Assignment - 5 - Content

- As expected, MONK-2 would be the most difficult dataset to classify; we need six attributes to build the decision tree.
- MONK-3 has 5% additional noise (misclassification) in the training set, so i originally thought that it is more difficult to learn than MONK-1, but the results show that MONK-1's error testing is much higher than MONK-3.
- All the error training is 0, which may mean overfitting.

Assignment - 5 - Content



Assignment - 5 - Content



Assignment - 5 - Content

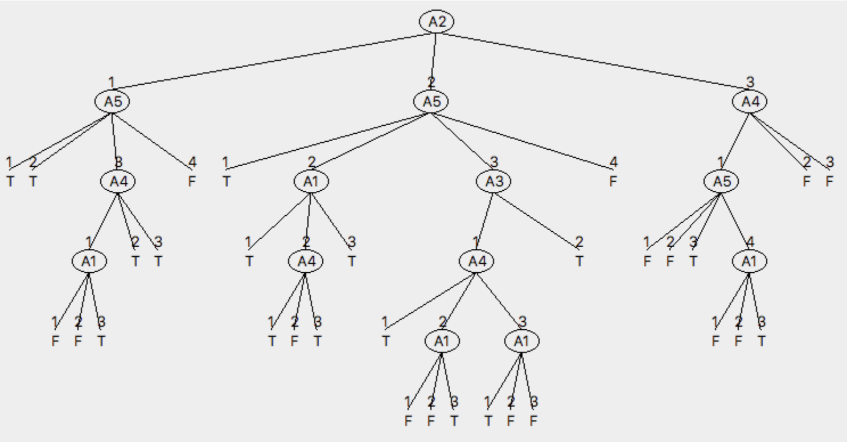


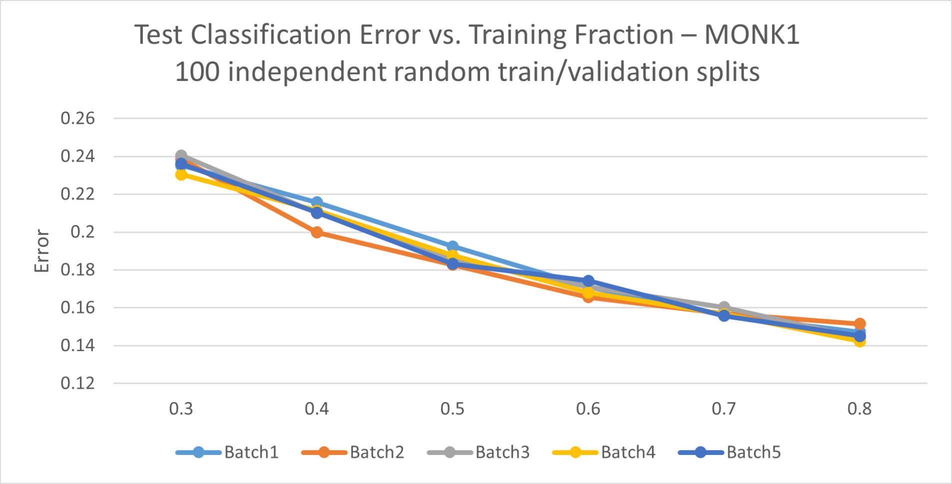
Table of Contents

Assignment - 6 - Content

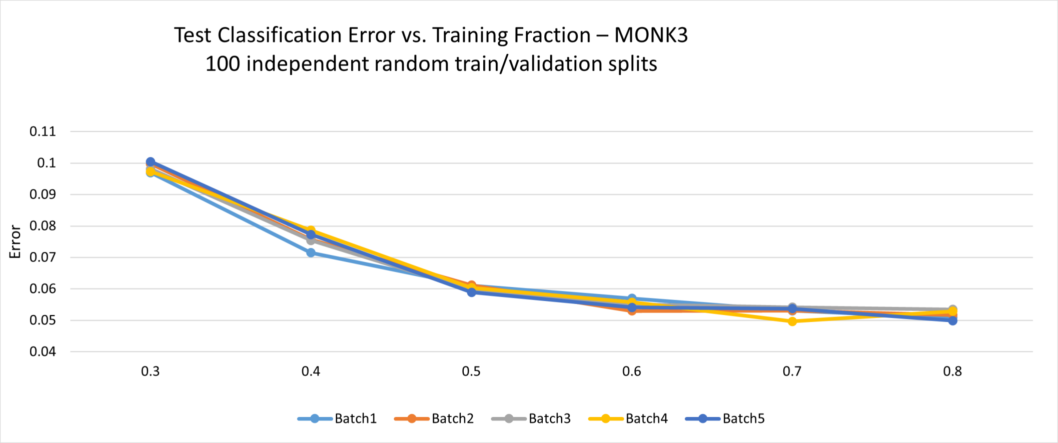
- If too many attributes are considered, the decision tree may grow very deep with many layers. This often leads to overfitting, where the model fits the training data almost perfectly but performs poorly on the test data, resulting in reduced generalization ability.
- With pruning, we cut some branches, reducing the complexity of the decision model, which improves its generalization performance.

Table of Contents

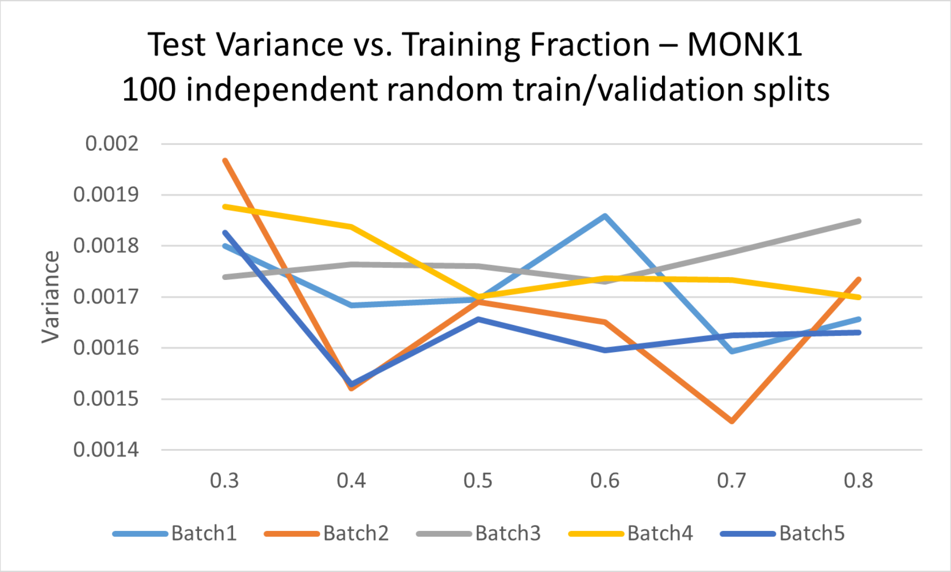
Assignment - 7 - Content



Assignment - 7 - Content



Assignment - 7 - Content



Assignment - 7 - Content